Computational Modeling of Narrative Texts, Films and Games

Reasoning about time and change

Rémi Ronfard, April 2016

Reasoning about time and change

- Things change over time
  - In the living room, Lisa picked up a newspaper and walked into the kitchen.
  - Kate set a book on the coffee table and left the living room. When she returned, the book was gone. Someone must have take it.
- Commonsense law of inertia = things stay the same unless affected by some event.
  - Jamie goes to the kitchen sink, puts the stopper in the drain, turns on the faucets and leaves.

Reasoning about time and change

- Default reasoning
  - Kimberley turns on the fan. It starts turning … or not (maybe it was not plugged in)
- Mental events
  - A hungry cat saw some food on a table. The cat jumped onto a chair.

Reasoning about time and change

- Prediction (temporal projection) from initial state and events to final state.
- Postdiction from events and final state to initial state.
- Abduction (explanation) from initial and final states to events.

Fluents and events

- Fluents are time-varying properties of the world
  - HoldsAt(f,t)
- Events - everything that happens in the world
  - Happens(e,t)
- Fluents can change values as a result of events

Non-Monotonic Logic

- In monotonic logic
  - A => C implies A and B => C
- Not so in temporal logic
  - event A => fluent C is true
  - event A then event B => fluent C not true
  - example : switch on the light then switch off the light

The Frame Problem

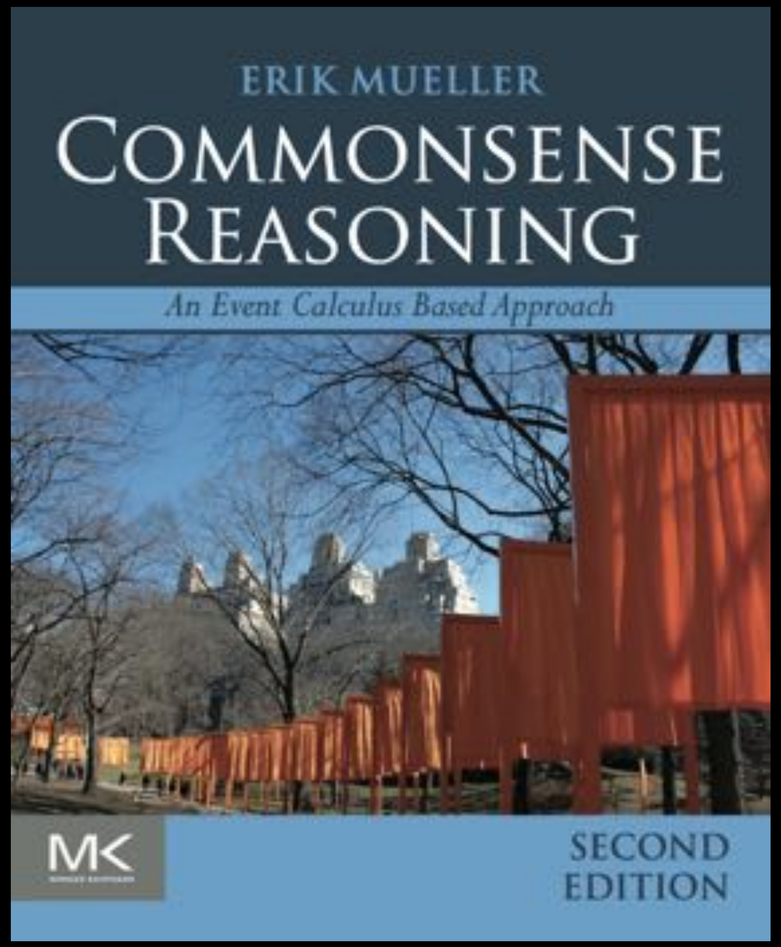
- Solving the Frame Problem
- A Mathematical Investigation of the Common Sense Law of Inertia
- By Murray Shanahan

Solving the frame problem

- Situation calculus : successor state actions
  - Give consequences of all events/actions
  - Assume everything else remains the same
- Event calculus : circumscription
- In both cases, requires a « domain theory »
  - How the real world changes as a result of events
  - How the fiction world changes as a result of events

Commonsense in stories

- Given a current situation (fluents) and a new event
  - How does the next situation (fluents) change ?
- Given a list of past situations (fluents) and a new telling of a past event
  - How should the situations (fluents) be revised ?



Discrete Event Calculus

- First-order logic representation of a story
- Fluents are time-varying properties of the world
  - HoldsAt(f,t)
- Events - everything that happens in the world
  - Happens(e,t)
- Fluents can change values as a result of events
  - Initiates(e,f,t) or Terminates(e,f,t)

Discrete Event Calculus

- ReleasedAt(f,t) : fluent is no longer subject to the commonsense law of inertia
  - the water level increases, the newspaper changes location, etc.
- Trajectory(f1,t1,f2,t2) : if f1 is initiated at t1 then f2 is true at t1 + t2
- Anti-trajectory(f1,t1,f2,t2) : if f1 is terminated at t1 then f2 is true at t1 + t2

Circumscription

- CIRC(Initiates(SwitchOn,LightOn,t) is
  - Initiates(e,f,t) IFF e = SwitchOn and f = LightOn
- CIRC(Terminates(SwitchOff,LightOn,t)) is
  - Terminates(e,f,t) IFF e = SwitchOff and f = LightOn

Discrete Event Calculus Axioms

- Axiom DEC1
  - StoppedIn(t1,f,t2) iff Happens(e,t) and t1 < t < t2 and Terminates(e,f,t)
- Axiom DEC2
  - StartedIn(t1,f,t2) iff Happens(e,t) and t1 < t < t2 and Initiates(e,f,t)

Discrete Event Calculus Axioms

Axiom DEC3

$$Happens(e, t_1) \wedge Initiates(e, f_1, t_1) \wedge 0 < t_2 \wedge$$
$$Trajectory(f_1, t_1, f_2, t_2) \wedge \neg StoppedIn(t_1, f_1, t_1 + t_2) \Rightarrow$$
$$HoldsAt(f_2, t_1 + t_2)$$

Discrete Event Calculus Axioms

Axiom DEC4

$$Happens(e, t_1) \wedge Terminates(e, f_1, t_1) \wedge 0 < t_2 \wedge$$
$$AntiTrajectory(f_1, t_1, f_2, t_2) \wedge \neg StartedIn(t_1, f_1, t_1 + t_2) \Rightarrow$$
$$HoldsAt(f_2, t_1 + t_2)$$

Discrete Event Calculus Axioms

- Axiom DEC5

$$HoldsAt(f,t) \wedge \neg ReleasedAt(f,t+1) \wedge$$
$$\neg \exists e\left(Happens(e,t) \wedge Terminates(e,f,t)\right) \Rightarrow$$
$$HoldeAt(f,t+1)$$

Discrete Event Calculus Axioms

- Axiom DEC6

$$\neg HoldsAt(f,t) \wedge \neg ReleasedAt(f,t+1) \wedge$$
$$\neg \exists e\left(Happens(e,t) \wedge Initiates(e,f,t)\right) \Rightarrow$$
$$\neg HoldeAt(f,t+1)$$

Discrete Event Calculus Axioms

- Axiom DEC7

$$ReleasedAt(f,t) \wedge$$
$$\neg \exists e\big(Happens(e,t) \wedge (Initiates(e,f,t) \vee Terminates(e,f,t))\big) \Rightarrow$$
$$ReleasedAt(f,t+1)$$

Discrete Event Calculus Axioms

- DEC8

$$\neg ReleasedAt(f,t) \wedge \neg \exists e\big(Happens(e,t) \wedge Releases(e,f,t)\big) \Rightarrow$$
$$\neg ReleasedAt(f,t+1)$$

Discrete Event Calculus Axioms

Axiom DEC9

If a fluent is initiated by some event that occurs at timepoint t, then the fluent is true at $t + 1$.

$$Happens(e,t) \wedge Initiates(e,f,t) \Rightarrow HoldsAt(f,t+1)$$

Discrete Event Calculus Axioms

Axiom DEC10

If a fluent is terminated by some event that occurs at timepoint t, then the fluent is false at $t + 1$.

$$Happens(e,t) \wedge Terminates(e,f,t) \Rightarrow \neg HoldsAt(f,t+1)$$

Discrete Event Calculus Axioms

Axiom DEC11

If a fluent is released by some event that occurs at timepoint t, then the fluent is released from the commonsense law of inertia at $t + 1$.

$$Happens(e,t) \wedge Releases(e,f,t) \Rightarrow ReleasedAt(f,t+1)$$

Discrete Event Calculus

- Axiom DEC 12

$$Happens(e,t) \wedge Initiates(e,f,t) \vee Terminates(e,f,t) \Rightarrow \neg ReleasedAt(f,t+1)$$

Domain Descriptions

- Effect axioms
  - Condition => Initiates(e,f,t) or Terminates(e,f,t)
- Fluent pre-conditions
  - HoldsAt(f,t) => Condition
  - NOT Condition => NOT HoldsAt(f,t)
- Action pre-conditions
  - Happens(e,t) => condition
  - NOT condition => NOT Happens(e,t)

Eric Mueller’s work on story understanding

- Build domain descriptions suitable for understanding « simple » stories
  - Stories about going to the restaurant
  - Stories about building a snowman
  - Stories about inviting a friend
- One difficulty is to build « generic » domain descriptions
- Another difficulty is to build domain description for fictional worlds

Example : The snowman by Raymond Briggs

- The Snowman is the tale of a boy who builds a snowman one winter's day. That night, at the stroke of twelve, the snowman comes to life.
- The first part of the story deals with the snowman's attempts to understand the appliances, toys and other bric-a-brac in the boy's house, all while keeping quiet enough not to wake the boy's parents. The two then venture back outside and go for a ride on a motorbike, disturbing many animals: pheasants, rabbits, a barn owl, a fox and a brown horse.
- In the second part of the story, the boy and the snowman take flight.
- The morning after the return journey the sun has come out and the boy wakes up to find the snowman has melted. As the credits play, the boy mourns the loss of his new friend.

Eric Mueller's analysis of the snowman story

- Source: story understanding through multi-representation model construction (2003)
- Manual annotation of story events in the discrete event calculus
- Build a complete model of the story as a suitable sequence of states (fluents) and events

The snowman story

PART 1 Hooray! It is snowing!

James gets dressed. He runs outside.

He makes a pile of snow. He makes it bigger and bigger.

He puts a big snowball on top. He adds a scarf and a hat,

an orange for a nose, coal for eyes and buttons.

There! What a fine snowman!

PART 2 It is nighttime. James sneaks downstairs.

He looks out the door. What does he see?

The snowman is moving!

James invites him in.

PART 3 The snowman has never been inside a house.

Hello, cat! Hello, lamp! Hello, paper towels!

The snowman takes James's hand.

They go up, up, up into the air!

They are flying! What a wonderful night!

PART 4 It is morning.

James jumps out of bed.

He runs downstairs.

He runs into the kitchen.

He runs outside. But the snowman has gone.

The snowman

- SNOWMAN1:
- This segment models the falling of individual snowflakes.

The snowman

- SNOWMAN2:
- Hooray!
- Happens(CryForJoy(James), 3)
- It is snowing!
- HoldsAt(Snowing(JamesOutside), 3)
- James gets dressed.
- Happens(GetDressed(James), 5)
- He runs outside.

The snowman

- Happens(WalkThroughDoor21(James, JamesFrontDoor1Fl),10)
- He makes a pile of snow.
- Happens(HoldSome(James, Snowball1, Snow1), 12)
- He makes it bigger and bigger.
- Happens(RollAlong(James, Snowball1, Snow1), 13)
- He puts a big snowball on top.
- Happens(PlaceOn(James, Snowball2, Snowball1), 17)

The snowman

- SNOWMAN3:
- This segment models James going into the house to get a scarf, hat, and orange.
- SNOWMAN4:
- He adds a scarf and a hat.
- Happens(PlaceOn(James, JamesScarf, Snowball2), 0)
- Happens(PlaceOn(James, JamesHat, Snowball2), 1)
- He adds an orange for a nose.
- Happens(PutInside(James, JamesOrange, Snowball2), 2)
- He adds coal for eyes and buttons.
- Happens(PutInside(James, JamesCoal, Snowball2), 4)
- There! What a fine snowman!

The snowman

- SNOWMAN5:
- It is nighttime.
- Nighttime(0)

The snowman

- James sneaks downstairs.
- Happens(WalkDownStaircase(James, JamesStaircase1To2), 1)
- He looks out the door.
- Happens(LookAt(James, Snowman), 4)
- What does he see?
- The snowman is moving!
- Happens(Move(Snowman), 5)

The snowman

- James invites him in.
- Happens(InviteIn(James, Snowman, JamesFoyer1Fl), 6)
- The snowman has never been inside a house.

The snowman

- SNOWMAN6:
- Hello, cat!
- Happens(Greet(Snowman, JamesCat), 0)
- Hello, lamp!
- Happens(Greet(Snowman, JamesLamp), 1)
- Hello, paper towels!
- Happens(Greet(Snowman, JamesPaperTowels), 2)
- The snowman takes James's hand.
- Happens(Hold(Snowman, JamesHand), 7)

The snowman

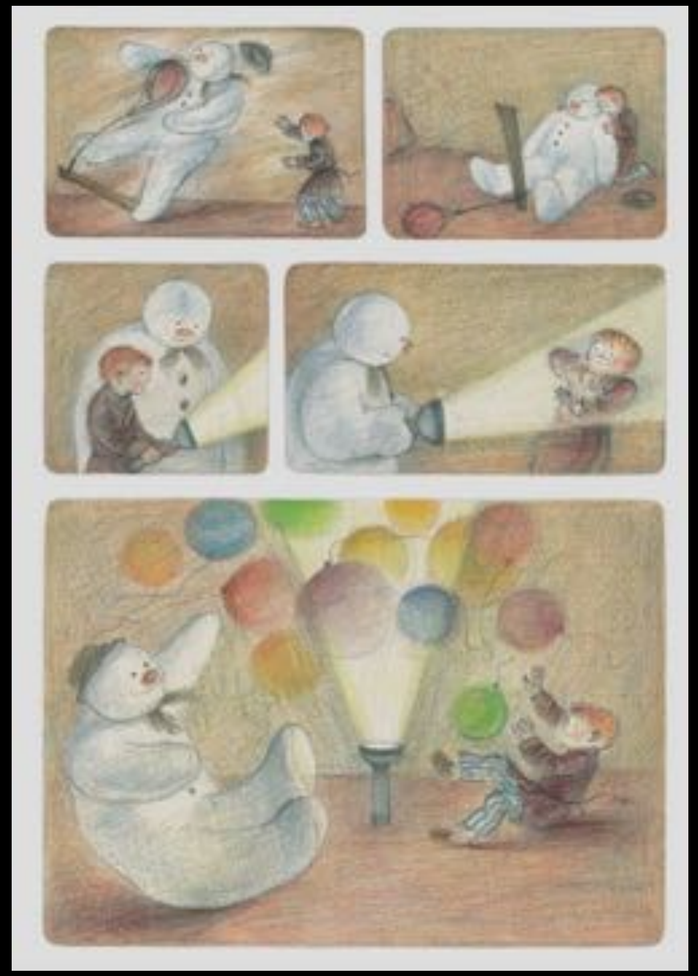
The snowman

The snowman

The snowman

The snowman

The snowman



The snowman

The snowman

The snowman

The snowman

The snowman

- SNOWMAN7:
- They go up, up, up into the air!
- Happens(StartFlyingFromTo(Snowman, JamesOutsideGround,JamesOutsideSky), 0)
- They are flying!
- HoldsAt(FlyingFromTo(Snowman, JamesOutsideGround, JamesOutsideSky), 1)
- What a wonderful night!

The snowman

The snowman

The snowman

The snowman

The snowman

The snowman

The snowman

The snowman

- SNOWMAN8:
- It is morning.
- Morning(0)

The snowman

- James jumps out of bed.
- Happens(RiseFrom(James, JamesBed), 1)
- He runs downstairs.
- Happens(WalkDownStaircase(James, JamesStaircase1To2), 4)
- He runs into the kitchen.
- He runs outside.
- Happens(WalkThroughDoor21(James, JamesFrontDoor1Fl),7)

The snowman

- But the snowman has gone.

Domain theories in The snowman

- CTime: clock time.
- ECTime: the event calculus model of time.
- Feeling: simple positive, neutral, and negative emotions, and positive, neutral, and negative attitudes toward objects.
- OMSpace: object-scale metric space, with falling and collisions.
- OTSpace: object-scale topological space.
- PlayNeed: the need to play, with a simple model of needs and intentions.
- RTSpace: room-scale topological space.
- Sleep: sleeping and body posture.
- Snow: snow and snow falling from the sky.
- SpeechAct: some simple speech acts.
- Vision: some simple aspects of vision.

Room Scale Topology axioms

Formula	English
Adjacent(location1, location2)	*location1* is adjacent to *location2*.
At(object, location)	*object* is at *location*.
BuildingOf(room) = building	The building of *room* is *building*.
DoorClose(actor, door)	*actor* closes *door*.
DoorIsOpen(door)	*door* is open.
DoorLock(actor, door)	*actor* locks *door*.
DoorOpen(actor, door)	*actor* opens *door*.
DoorUnlocked(door)	*door* is unlocked.
DoorUnlock(actor, door)	*actor* unlocks *door*.
Floor(room) = storey	The floor of *room* is *storey*.
GroundOf(outside) = ground	The ground of *outside* is *ground*.
LookOutOnto(room) = outside	*room* looks out onto *outside*.
NearPortal(object, portal)	*object* is at a location that has *portal*.
Side1(portal) = location	Side one of *portal* is *location*.
Side2(portal) = location	Side two of *portal* is *location*.
SkyOf(outside) = sky	The sky of *outside* is *sky*.
WalkDownStairwell(actor, stairwell)	*actor* walks down *stairwell*.
WalkThroughDoor1(actor, door)	*actor* walks through side one of *door*.
WalkThroughDoor2(actor, door)	*actor* walks through side two of *door*.
WalkUpStairwell(actor, stairwell)	*actor* walks up *stairwell*.

Domain axioms for the snowman story

Axiom 1.

$$HoldsAt(At(object, location1), time) \wedge HoldsAt(At(object, location2), time) \Rightarrow location1 = location2$$

Axiom 2.

$$HoldsAt(NearPortal(object, portal), time) \Leftrightarrow \exists\, location\ (Side1(portal) = location \vee Side2(portal) = location) \wedge HoldsAt(At(object, location), time)$$

Axiom 3.

$$Happens(DoorUnlock(actor, door), time) \Rightarrow HoldsAt(Awake(actor), time) \wedge \neg HoldsAt(DoorUnlocked(door), time) \wedge HoldsAt(NearPortal(actor, door), time)$$

Axiom 4.

$$Initiates(DoorUnlock(actor, door), DoorUnlocked(door), time)$$

Domain axioms for the snowman story

Axiom 5.
$HoldsAt(DoorIsOpen(door), time) \Rightarrow$
$HoldsAt(DoorUnlocked(door), time)$

Axiom 6.
$Happens(DoorOpen(actor, door), time) \Rightarrow$
$HoldsAt(Awake(actor), time) \wedge$
$\neg HoldsAt(DoorIsOpen(door), time) \wedge$
$HoldsAt(DoorUnlocked(door), time) \wedge$
$HoldsAt(NearPortal(actor, door), time)$

Axiom 7.
$Initiates(DoorOpen(actor, door), DoorIsOpen(door), time)$

Axiom 8.
$Happens(WalkThroughDoor12(actor, door), time) \Rightarrow$
$HoldsAt(Awake(actor), time) \wedge$
$HoldsAt(Standing(actor), time) \wedge$
$HoldsAt(DoorIsOpen(door), time) \wedge$
$HoldsAt(At(actor, Side1(door)), time)$

Domain axioms for the snowman story

Axiom 9.
$Happens(WalkThroughDoor21(actor, door), time) \Rightarrow$
$HoldsAt(Awake(actor), time) \wedge$
$HoldsAt(Standing(actor), time) \wedge$
$HoldsAt(DoorIsOpen(door), time) \wedge$
$HoldsAt(At(actor, Side2(door)), time)$

Axiom 10.
$Side2(door) = location \Rightarrow$
$Initiates(WalkThroughDoor12(actor, door),$
$At(actor, location), time)$

Axiom 11.
$Side1(door) = location \Rightarrow$
$Initiates(WalkThroughDoor21(actor, door),$
$At(actor, location), time)$

Axiom 12.
$Side1(door) = location \Rightarrow$
$Terminates(WalkThroughDoor12(actor, door),$
$At(actor, location), time)$

Domain axioms for the snowman story

Axiom 13.
$Side2(door) = location \Rightarrow$
$Terminates(WalkThroughDoor21(actor, door),$
$At(actor, location), time)$

Axiom 14.
$HoldsAt(At(actor, outside), time) \Rightarrow$
$HoldsAt(Dressed(actor), time)$

Axiom 15.
$Adjacent(location1, location2) \Leftrightarrow$
$\exists\, portal\ (Side1(portal) = location1 \wedge$
$Side2(portal) = location2) \vee$
$(Side2(portal) = location1 \wedge$
$Side1(portal) = location2)$

Axiom 16.
$GroundOf(outside) = ground \Rightarrow$
$HoldsAt(At(ground, outside), time)$

Axiom 17.
$SkyOf(outside) = sky \Rightarrow$
$HoldsAt(At(sky, outside), time)$

Model of segment 2

1
+Snowing(JamesOutside)
Happens(WakeUp(James), 1)
2
Event occurrences are shown at the end of each time point. Only changes in what fluents hold from one time point to the next are shown. Thus after the WakeUp event occurs above, James is no longer asleep and he is awake:
-Asleep(James)
-Sleep0(James)
+Awake(James)
+Sleep1(James)
An axiom in the Feeling representation triggers this event in response to the snow:
Happens(BecomeHappy(James), 2)

An axiom in the PlayNeed representation triggers this event in response to the snow:

Happens(IntendToPlay(James, JamesOutside), 2)
3
-Calm(James)
-HungryToPlay(James)
+Happy(James)
+IntentionToPlay(James, JamesOutside)
Happens(CryForJoy(James), 3)
4
Happens(RiseFrom(James, JamesBed), 4)

Model of segment 2

5
-*LyingOn(James, JamesBed)*
-*Lying(James)*
-*Sleep1(James)*
+*Sleep2(James)*
+*Standing(James)*
Happens(GetDressed(James), 5)
6
-*Sleep2(James)*
+*Dressed(James)*
+*Sleep3(James)*
Happens(WalkThroughDoor12(James, JamesDoor2Fl), 6)

7
James was in his bedroom from time points 0 to 6 inclusive. After he walks through the bedroom door above, he is no longer in his bedroom:
-*At(James, JamesBedroom2Fl)*
+*At(James, JamesHallway2Fl)*
+*NearPortal(James, JamesStaircase1To2)*
Happens(WalkDownStaircase(James, JamesStaircase1To2), 7)
8
-*At(James, JamesHallway2Fl)*
-*NearPortal(James, JamesDoor2Fl)*
+*At(James, JamesFoyer1Fl)*
+*NearPortal(James, JamesFrontDoor1Fl)*
+*NearPortal(James, JamesKitchenDoor1Fl)*
Happens(DoorUnlock(James, JamesFrontDoor1Fl), 8)

Model of segment 2

9
+DoorUnlocked(JamesFrontDoor1Fl)
Happens(DoorOpen(James, JamesFrontDoor1Fl), 9)
10
+DoorIsOpen(JamesFrontDoor1Fl)
Happens(WalkThroughDoor21(James, JamesFrontDoor1Fl),
10)
11
-At(James, JamesFoyer1Fl)
-NearPortal(James, JamesKitchenDoor1Fl)
-NearPortal(James, JamesStaircase1To2)

Optional intermediate fluents fix the time point at which James acts on his intention to play, thereby reducing the number of models:
+ActOnIntentionToPlay(James, JamesOutside)
+At(James, JamesOutside)
Happens(Play(James, JamesOutside), 11)
12
-ActOnIntentionToPlay(James, JamesOutside)
-IntentionToPlay(James, JamesOutside)
+SatiatedFromPlay(James)
Happens(HoldSome(James, Snowball1, Snow1), 12)
13
+Holding(James, Snowball1)
Happens(RollAlong(James, Snowball1, Snow1), 13)

Model of segment 2

14
-*Diameter*(*Snowball1*, 1)
+*Diameter*(*Snowball1*, 2)
Happens(*LetGoOf*(*James*, *Snowball1*), 14)

15
-*Holding*(*James*, *Snowball1*)
Happens(*HoldSome*(*James*, *Snowball2*, *Snow1*), 15)
16
+*Holding*(*James*, *Snowball2*)
Happens(*RollAlong*(*James*, *Snowball2*, *Snow1*), 16)
17
-*Diameter*(*Snowball2*, 1)
+*Diameter*(*Snowball2*, 2)
Happens(*PlaceOn*(*James*, *Snowball2*, *Snowball1*), 17)
18
-*Holding*(*James*, *Snowball2*)
+*On*(*Snowball2*, *Snowball1*)

Conclusions and directions for future work

- Story understanding as abduction
  - Charniak, Model of children's story comprehension, MIT, 1972.
  - Hobbs et al. Interpretation as abduction, AIJ, 1993.
- Given narrated events (and domain theories), add implicit events until a consistent model can be computed
- Consistent model can be used to answer questions
  - Semantic parsing for input
  - Natural language generation for output


The End